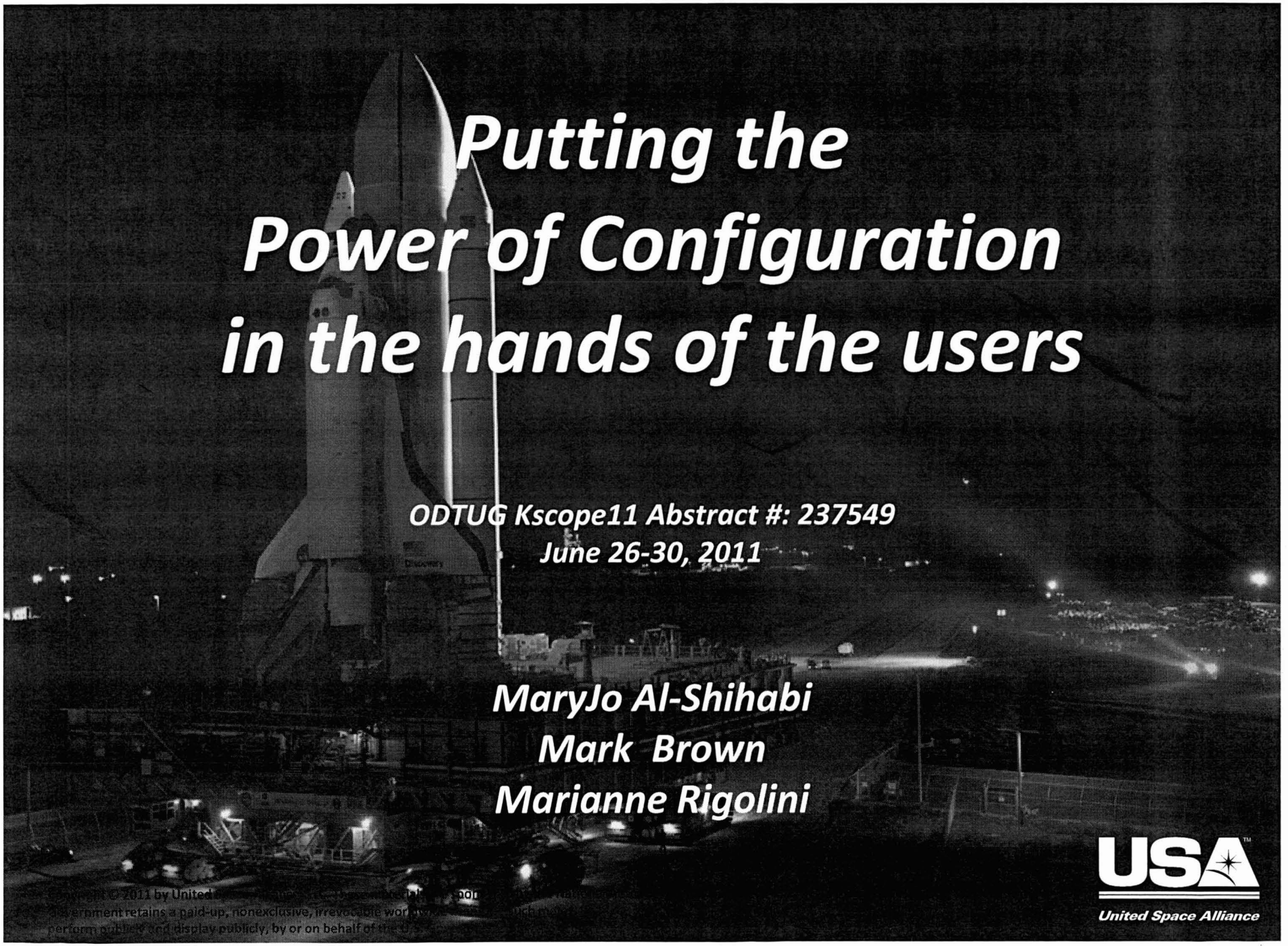


Putting the Power of Configuration in the hands of the users

ODTUG Kscope11 Abstract #: 237549
June 26-30, 2011

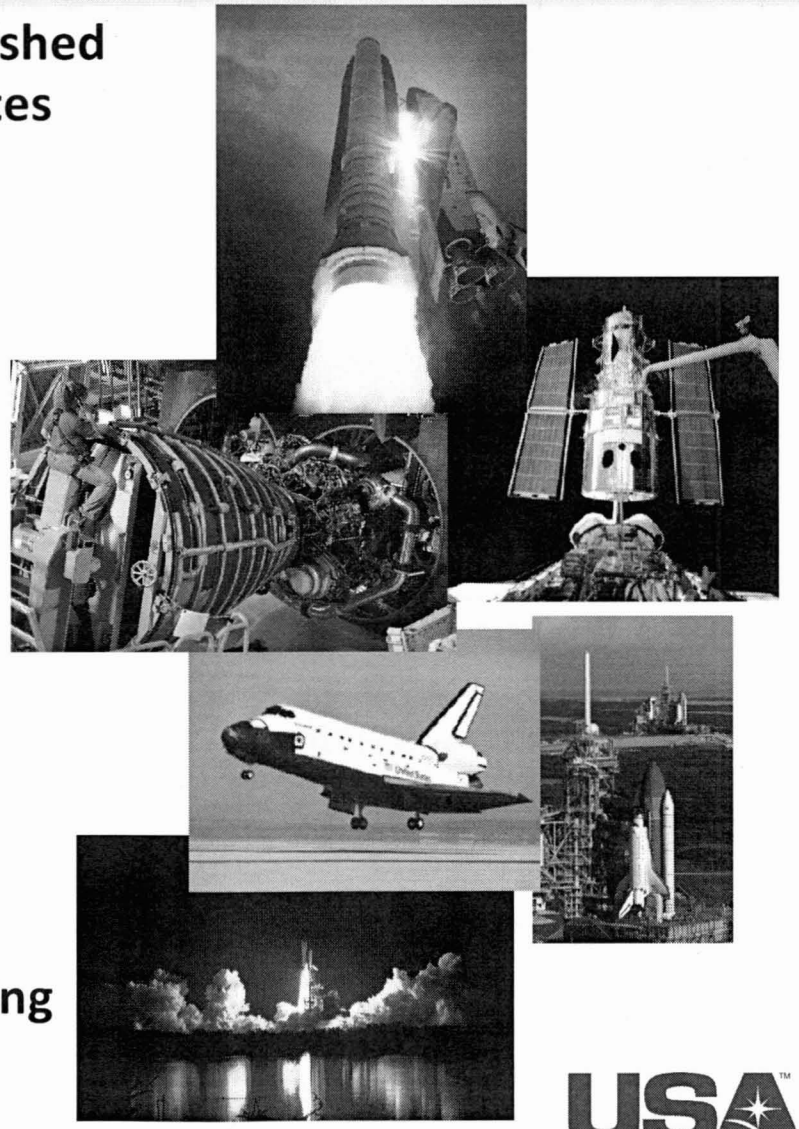
MaryJo Al-Shihabi
Mark Brown
Marianne Rigolini

Copyright © 2011 by United
Government retains a paid-up, nonexclusive, irrevocable wor
perform publicly and display publicly, by or on behalf of the U

USA™
United Space Alliance

United Space Alliance, LLC (USA)

- Headquartered in Houston, USA was established in 1995 with 8,800 employees working at sites in Texas, Florida, and Alabama
- Our mission is to provide safe, high-quality, best-value space operations, services and technologies to our customers
- World leader in Space Operations
  - Space Flight Mission Operations
  - Space Systems Integrated Logistics
  - Space Systems Integration and Program Management
  - Space Systems Ground Operations and Processing
  - Space Systems Flight Software
- Successfully completed end-to-end processing for 56 Space Shuttle Missions



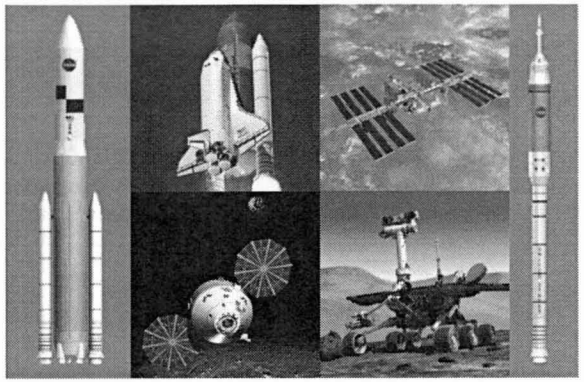
USA™
United Space Alliance

Background

- USA's Space Flight Operations Contract represented the integration of 12 NASA contracts that were previously managed by dozens of independent and diverse contractors
  - Goal was to reduce the overall cost of human space flight while maintaining the most demanding standards for safety and mission success
- In support of this goal, a project team was chartered to replace 18 legacy Space Shuttle nonconformance processes and systems with one fully integrated system
  - Problem Reporting and Corrective Action (PRACA) processes provide a closed-loop system for the identification, disposition, resolution, closure, and reporting of all Space Shuttle hardware/software problems
  - PRACA processes are integrated throughout the Space Shuttle organizational processes and are critical to assuring a safe and successful program

Challenge

- **Primary Project Objectives**
  - **Develop a fully integrated system that provides automated workflow with electronic signatures**
  - **Support multiple NASA programs and contracts with a single "system" architecture**
  - **Define standard processes, implement best practices, and minimize process variations**
  - **Reduce system operating costs and supportability risks**

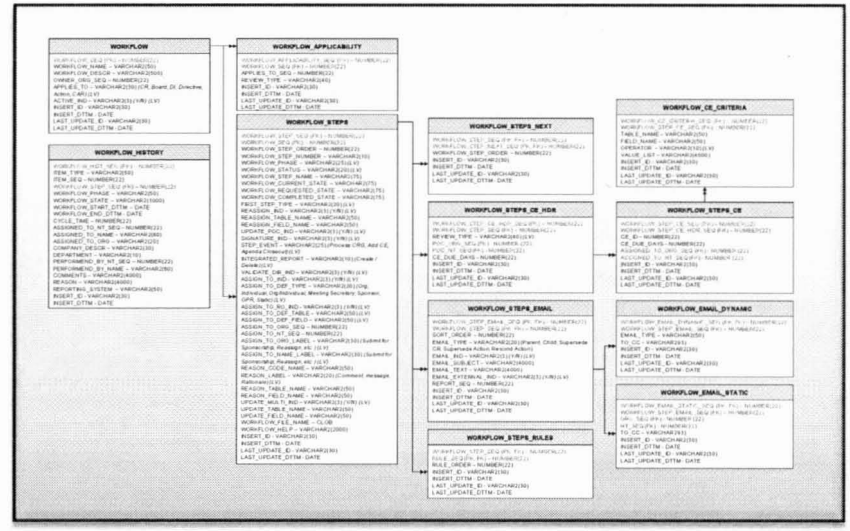


How do you integrate multiple diverse systems when the business and system requirements are not approved or even fully defined?

How do you get multiple customers to buy-in to a common process and system when historically they had complete independence and control?

Solution – Design for Flexibility

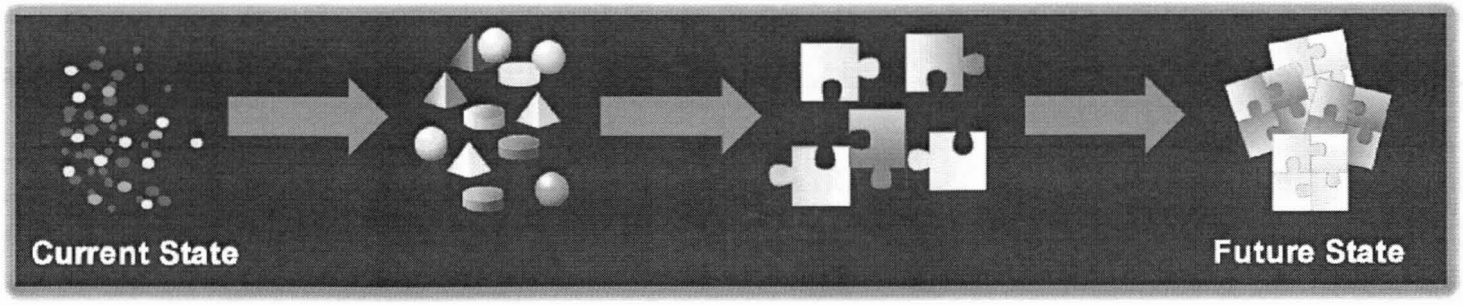
- Develop a configurable system that can be tailored by the user community as new business processes are defined and customer buy-in evolves
- A configurable system allows many functions to be data driven, allowing users to configure and implement changes without software code impacts
  - Data Integrity Drivers – Value Lists, Validation Rules, and Security
  - Automated Workflow, Electronic Signatures, and E-Mail Notifications
  - Data Entry Screens
  - Change Tracking and Audit
  - In Box and Assignment Lists
  - Page and Field Level Help
  - Search and Reporting
  - System Interfaces
  - Fail-Over Architecture



United Space Alliance

Advantages

- Configurable systems allow development to proceed without a finalized business process
- Highly flexible and robust design that can support Enterprise, Contract, Program, and/or Organizational processing
- Users can control process change implementations more efficiently
- Software code changes are minimized\eliminated
  - Increased reliability and reduced downtime
  - Reduced testing and verification
  - Reduced system maintenance cost
- Reusable Software Code



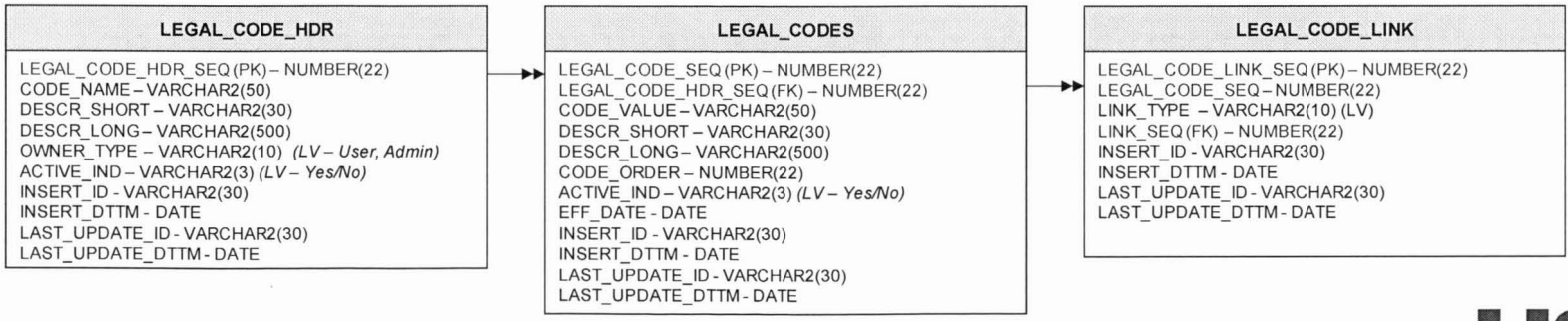
Configuration Design Approach

- **Design for Flexibility and Growth**
  - **Determine the primary configuration drivers (i.e. Contract, Organization, and/or Process Variations)**
  - **Evaluate requirements to identify items that would benefit from a configurable design**
    - **High Risk requirements – details are not fully defined**
    - **Unique functionality required to support different contracts, organizations or processes**
    - **Areas where you anticipate change as the business process evolves**
    - **Reusable functions**
- **Consider hard-coding functions that will be applied across the application and flexibility is not required or necessarily desired such as core business requirements**

Cost benefits maximized with a Balanced approach

Example 1 – LOV Design

- **Legal Values (or LOV) represents a list of valid values a user may select from a pre-defined list**
- **LOV maintained in a central table and dynamically derived**
  - **LEGAL_CODE_HDR– Groups the different legal value lists**
    - **CODE_NAME - Equates to the data field on a screen that is linked to the LOV**
    - **OWNER_TYPE – Defines who can update\maintain LOV**
  - **LEGAL_CODES – Contain the actual individual values**
    - **ACTIVE_IND – Allows the user to deactivate a value without removing it from the list**
    - **EFF_DATE – Identifies when the value became effective**
  - **LEGAL_CODE_LINK – Allows values to be filtered based on another value\field**
    - **LINK_TYPE and LINK_SEQ – Identifies the source field for the sub-filtering**



Example 1 – LOV Configuration

Code Name : CATEGORY

Order	Value	Value Short Description	Value Active	Owner Type	Applies To Forms	Applies To Boards
10	Mission	Space Shuttle Mission Number	Yes	Form Type	SSP CR SSP Closeout CR	
20	Equipment	Equipment end-item or hardware serial number	Yes	Form Type	SSP CR SSP Closeout CR	
30	Facility	Manufacturing\processing building or work area	Yes	Form Type	SSP CR SSP Closeout CR	
40	Non-Flight Specific	Not associated with any specific flight, end-item, or facility	Yes	Form Type	SSP CR SSP Closeout CR	
50	Not Applicable		Yes	Form Type	SSP Closeout CR	

Code Name : CE E-Mail Recipient

Order	Value	Value Short Description	Value Active	Owner Type	Applies To Forms	Applies To Boards
10	Assigned To		Yes	Workflow		
20	CE Evaluator		Yes	Workflow		
30	CE Org Evaluators		Yes	Workflow		
40	CE Approver		Yes	Workflow		
50	CE POC		Yes	Workflow		
60	CR Initiator		Yes	Workflow		
70	CR OPR		Yes	Workflow		
			Yes	Workflow		

Code Name | Values

Create/Edit Legal Value Type — Cancel | Save

Code Name: CATEGORY Type: Form Type Active: Yes

Short Description: Effectivity category

20 of 30

Long Description: Defines the mission, equipment, facility, etc affected and/or impacted by the change request

92 of 500

Audit Information

Inserted By/On: USAFL\FLEMMIMJ 06/04/2009 07:35:48 AM

Last Updated By/On: USATX\BROWNMA 04/25/2011 02:50:38 PM

Code Name | Values

Create/Update Legal Value — Add Another Value | Cancel | Save

Value: Mission Order: 10 Active: Yes Effective: 06/04/2009

Short Description: Space Shuttle Mission Number

28 of 75

Long Description: Defines Effectivity by mission or STS cycle

43 of 500

Applies to Forms

Available: SSP ICB CR

Selected: SSP CR, SSP Closeout CR

Values are filtered by Form Type

Audit Information

Inserted By/On: USAFL\FLEMMIMJ 06/04/2009 07:36:05 AM

Last Updated By/On: USATX\BROWNMA 04/25/2011 02:42:58 PM

United Space Alliance

Example 1 – APEX LOV Definition

List of Values

Named LOV: - Select Named LOV -

Display Extra Values: No

Display Null Value: Yes

Null Display Value: - Select - Null Return Value: %null%

Cascading LOV Parent Item(s):

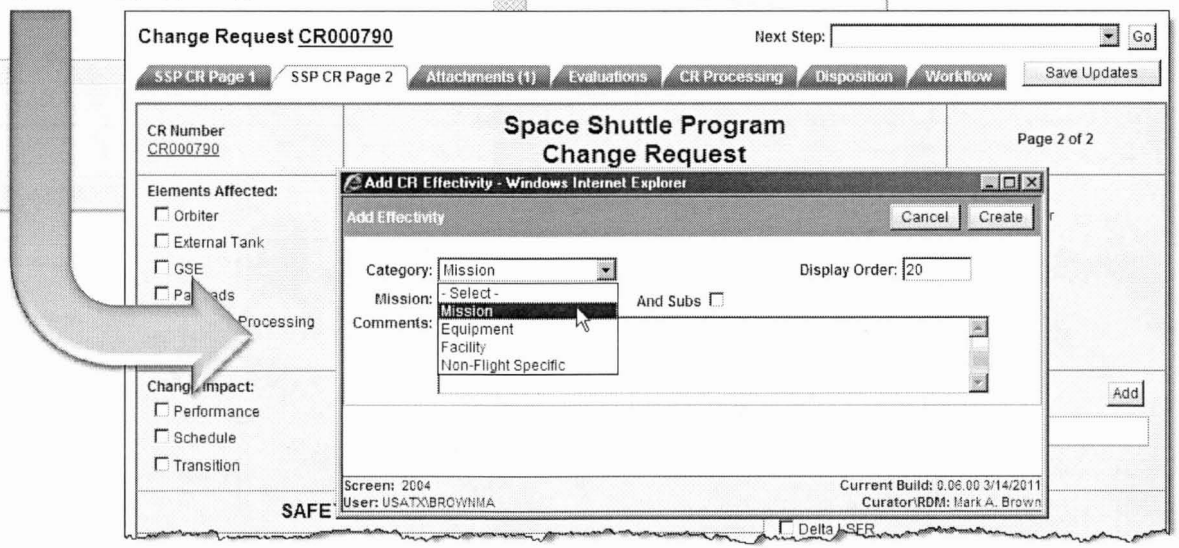
List of values definition

```
SELECT CODE_VALUE "D", CODE_VALUE "R"
  FROM CMST_LEGAL_CODES_ACTIVE_VW V, cmst_legal_code_link L WHERE CODE_NAME
= 'CATEGORY' and v.legal_code_seq=l.legal_code_seq
       and L.link_type='CR_FORM' and L.link_seq=:P2004_CR_FORM_SEQ
ORDER BY CODE_ORDER
```

Create or edit static List of Values

Create Dynamic List of Values

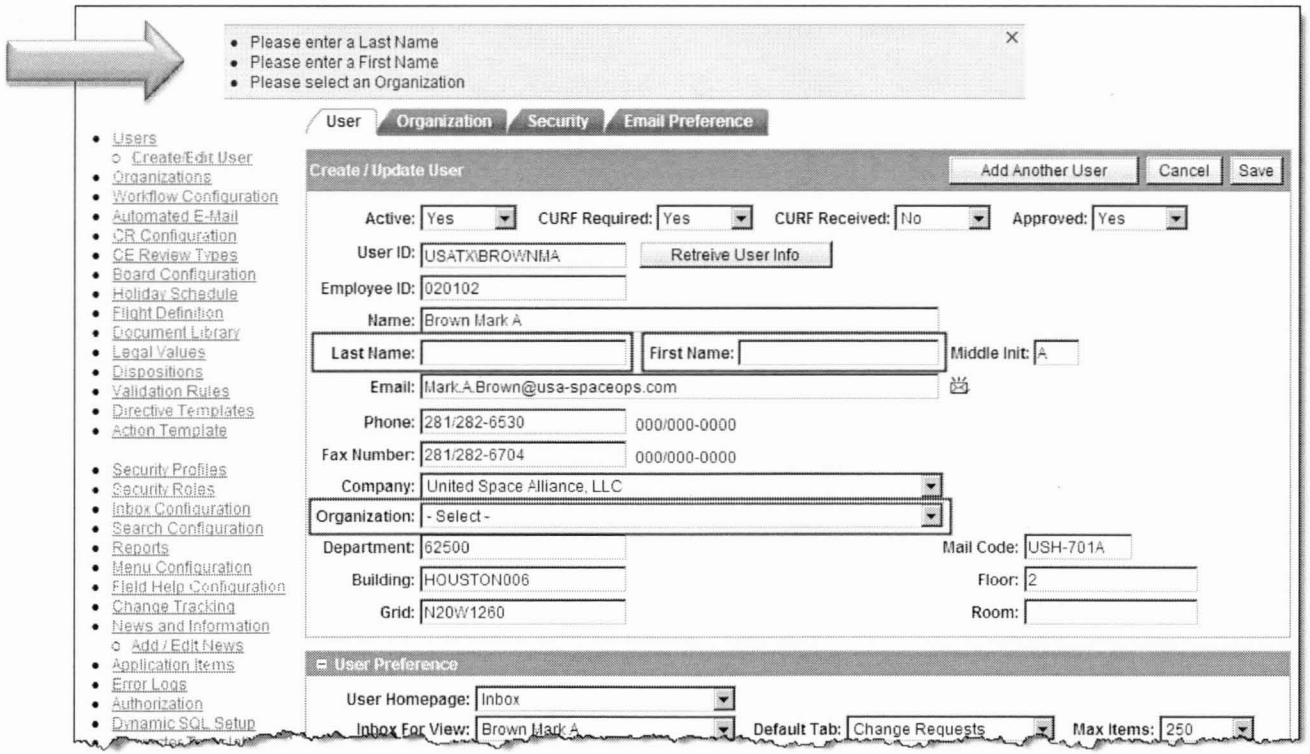
List of Values Examples



United Space Alliance

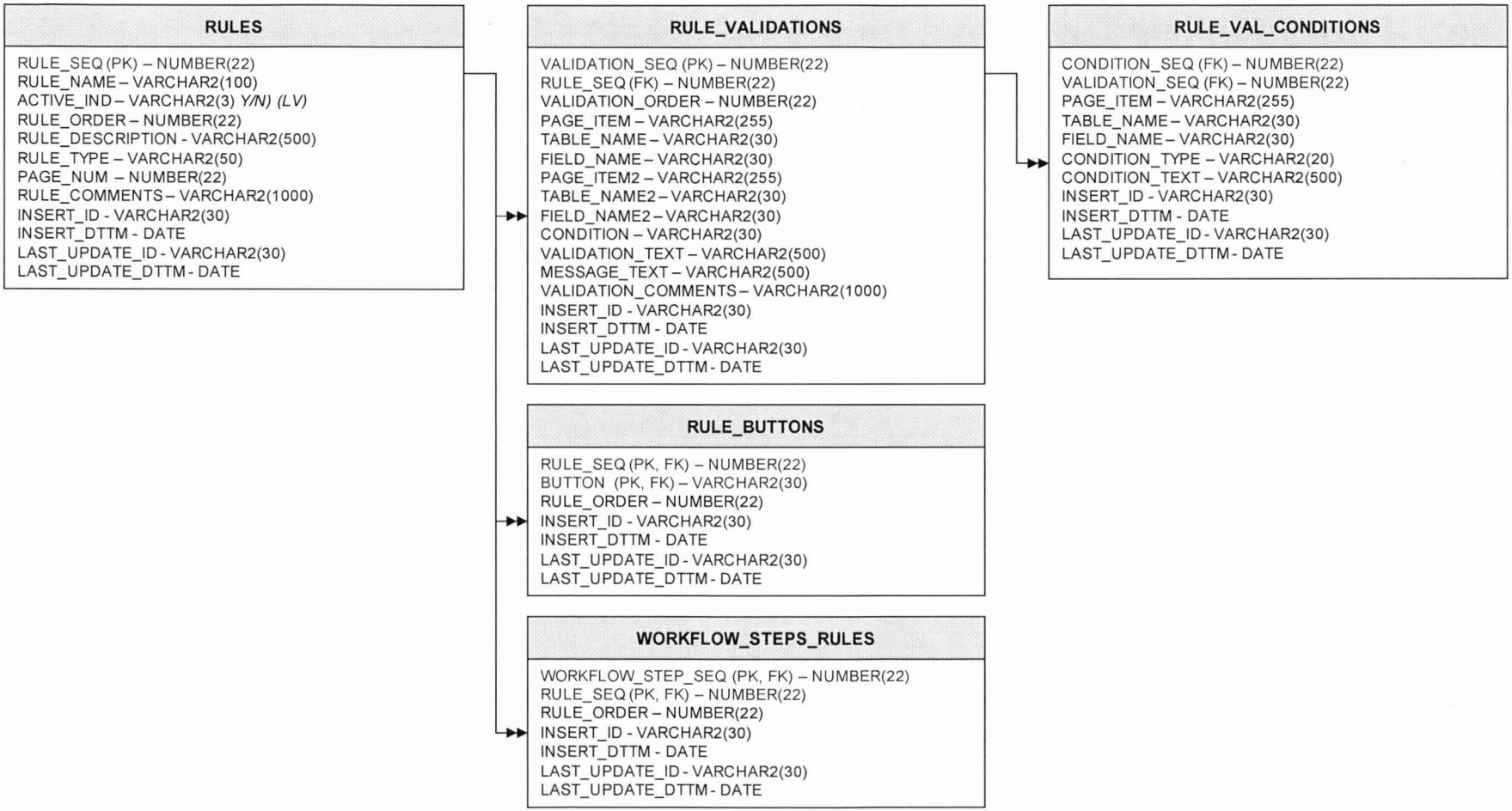
Example 2 – Validation Rules

- Validation Rules are used to ensure compliance with customer, company, organizational, and system data requirements
- Validation Rule configuration design generates an SQL statement that is evaluated when a Button is selected or a Workflow step is processed
- System will not commit changes until all validation errors are resolved



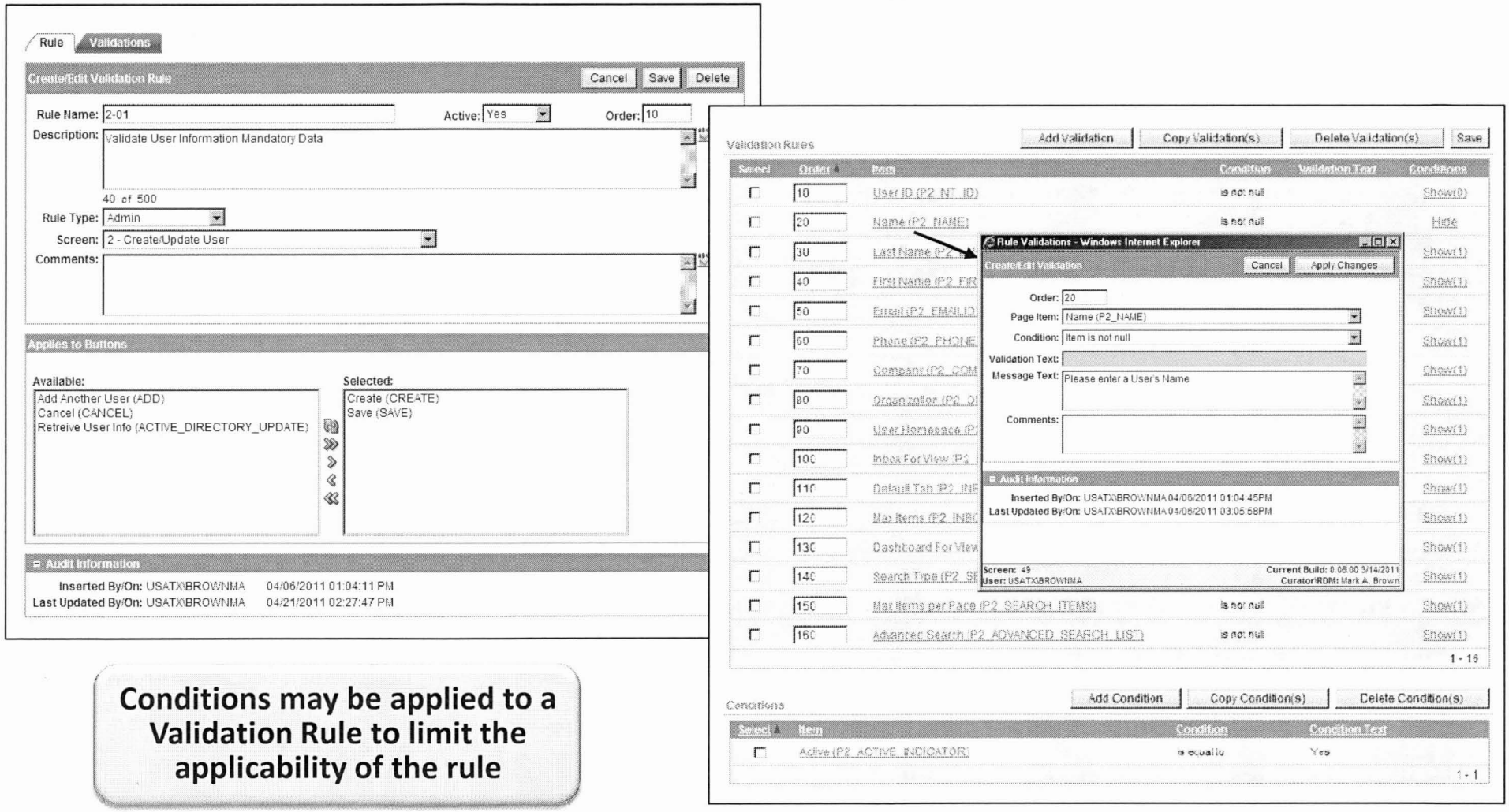
United Space Alliance

Example 2 – Validation Rule Design



USA™
United Space Alliance

Example 2 – Validation Rule Configuration



Conditions may be applied to a Validation Rule to limit the applicability of the rule

United Space Alliance

Example 2 – APEX Validation Function

Page Validation: **1 of 1** Name: **Get Screen Error Messages** Cancel Delete Apply Changes

Show All | Validation | Error Message | Conditions | Security | Configuration | Comments

Validation

Page: 2 Create/Update User

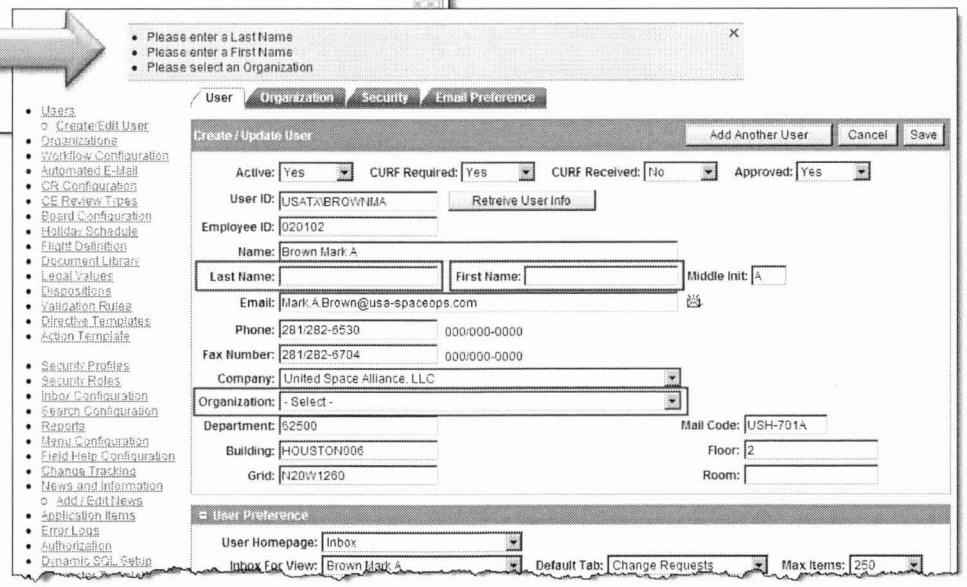
* Name: Get Screen Error Messages

* Sequence: 40

Type: Function Returning Error Text

* Validation Expression 1

```
BEGIN
  RETURN CMST_VALIDATION_PACK.GET_ERROR_MESSAGES(null,:APP_PAGE_ID, :REQUEST,null, null);
END;
```



- **Function (GET_ERROR_MESSAGES) processes each configured validation rule and identifies errors**
- **Errors are combined into a single message and displayed on the corresponding screen header**
- **System will not commit changes until all validation errors are resolved**

United Space Alliance

Example 2 – APEX Validation Behind the Scenes

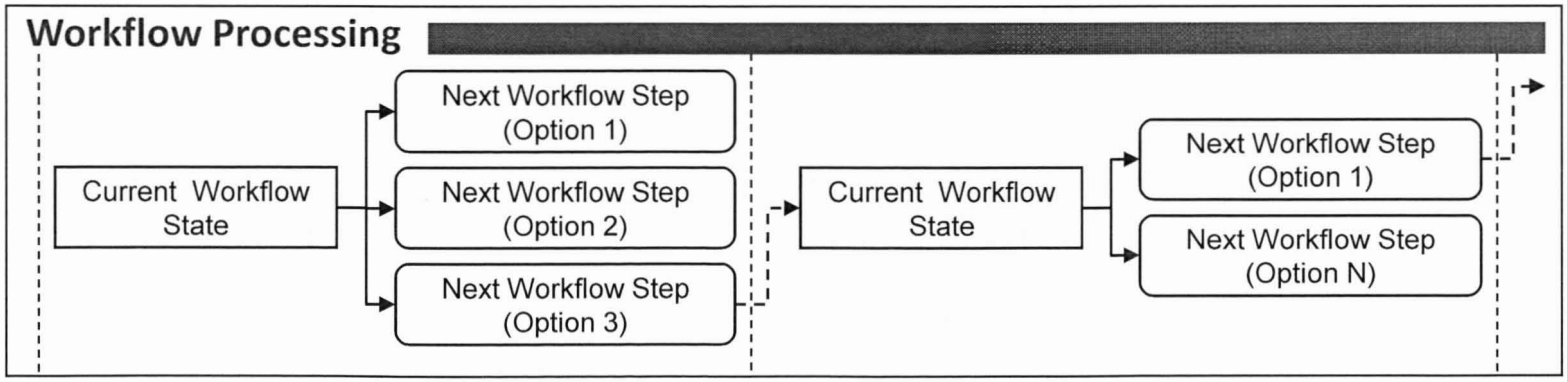
```
FOR validations IN (SELECT V.*  FROM CMST_RULES R, CMST_RULE_VALIDATIONS V, CMST_RULE_BUTTONS RB
      WHERE R.RULE_SEQ   = RB.RULE_SEQ  AND R.PAGE_NUM   = p_page_number AND RB.BUTTON    = p_request AND R.RULE_SEQ   = V.RULE_SEQ
            AND V.PAGE_ITEM IS NOT NULL AND NVL(R.ACTIVE_IND,'No') = 'Yes' AND (v_cr_form_seq IS NULL OR v_cr_form_seq
               IN (SELECT CR_FORM_SEQ FROM CMST_RULE_CR_FORMS WHERE RULE_SEQ = R.RULE_SEQ)) ORDER BY R.RULE_ORDER, V.VALIDATION_ORDER)
    LOOP
      v_run_val := TRUE;
      --CHECK ALL CONDITIONS TO SEE IF THE VALIDATION SHOULD RUN
      FOR condition IN (SELECT PAGE_ITEM, TABLE_NAME, FIELD_NAME, CONDITION_TYPE, CONDITION_TEXT
                          FROM CMST_RULE_VAL_CONDITIONS
                         WHERE VALIDATION_SEQ = validations.VALIDATION_SEQ
                           AND PAGE_ITEM IS NOT NULL)
      LOOP
        v_sql := 'SELECT 1 FROM DUAL WHERE ' || CMST_SQL_PACK.GET_SQL_COMPARE_TEXT(GET_DATATYPE(condition.CONDITION_TEXT)
                    ,GET_PAGE_ITEM_QUERY_TEXT(condition.PAGE_ITEM),condition.CONDITION_TYPE, condition.CONDITION_TEXT);
        BEGIN
          execute immediate v_sql INTO v_count;
        EXCEPTION
          WHEN NO_DATA_FOUND THEN v_count := 0;
          WHEN OTHERS THEN v_count := 1;
        END;
        IF v_count = 0 THEN v_run_val := FALSE;
          EXIT; --NO REASON TO GO ON, EXIT THE LOOP AND DON'T RUN THE VALIDATION
        END IF;
      END LOOP;
      IF v_run_val THEN
        IF UPPER(validations.VALIDATION_TEXT) = 'SYSDATE' THEN
          v_sql := 'SELECT 1 FROM DUAL WHERE TO_DATE(V(''' || validations.PAGE_ITEM || '''),''MM/DD/YYYY'') '  || validations.CONDITION ||
' TRUNC(' | validations.VALIDATION_TEXT || ')';
        ELSE
          IF validations.PAGE_ITEM2 IS NOT NULL THEN
            v_sql := 'SELECT 1 FROM DUAL WHERE '
                     ||
CMST_SQL_PACK.GET_SQL_COMPARE_TEXT(GET_DATATYPE(validations.VALIDATION_TEXT),GET_PAGE_ITEM_QUERY_TEXT(validations.PAGE_ITEM)
                      validations.CONDITION, GET_PAGE_ITEM_QUERY_TEXT(validations.PAGE_ITEM2));
          ELSE
            v_sql := 'SELECT 1 FROM DUAL WHERE ' || CMST_SQL_PACK.GET_
                     , GET_PAGE_ITEM_QUERY_TEXT(validations.PAGE_IT
          END IF;
        END IF;
        BEGIN
          execute immediate v_sql into v_count;
        EXCEPTION
          WHEN OTHERS THEN v_count := 0;
        END;
        IF v_count = 0 THEN v_errors := v_errors || validations.MESS
        END IF;
      END IF;
    END LOOP;
```

Loop through Rules to generate SQL

```
SELECT 1 FROM DUAL WHERE V('P2_NT_ID') IS NOT NULL
SELECT 1 FROM DUAL WHERE V('P2_NAME') IS NOT NULL
SELECT 1 FROM DUAL WHERE V('P2_LAST_NAME') IS NOT NULL
SELECT 1 FROM DUAL WHERE V('P2_FIRST_NAME') IS NOT NULL
SELECT 1 FROM DUAL WHERE V('P2_EMAILID') IS NOT NULL
SELECT 1 FROM DUAL WHERE V('P2_PHONE') IS NOT NULL
SELECT 1 FROM DUAL WHERE V('P2_COMPANY_DESCR') IS NOT NULL
```

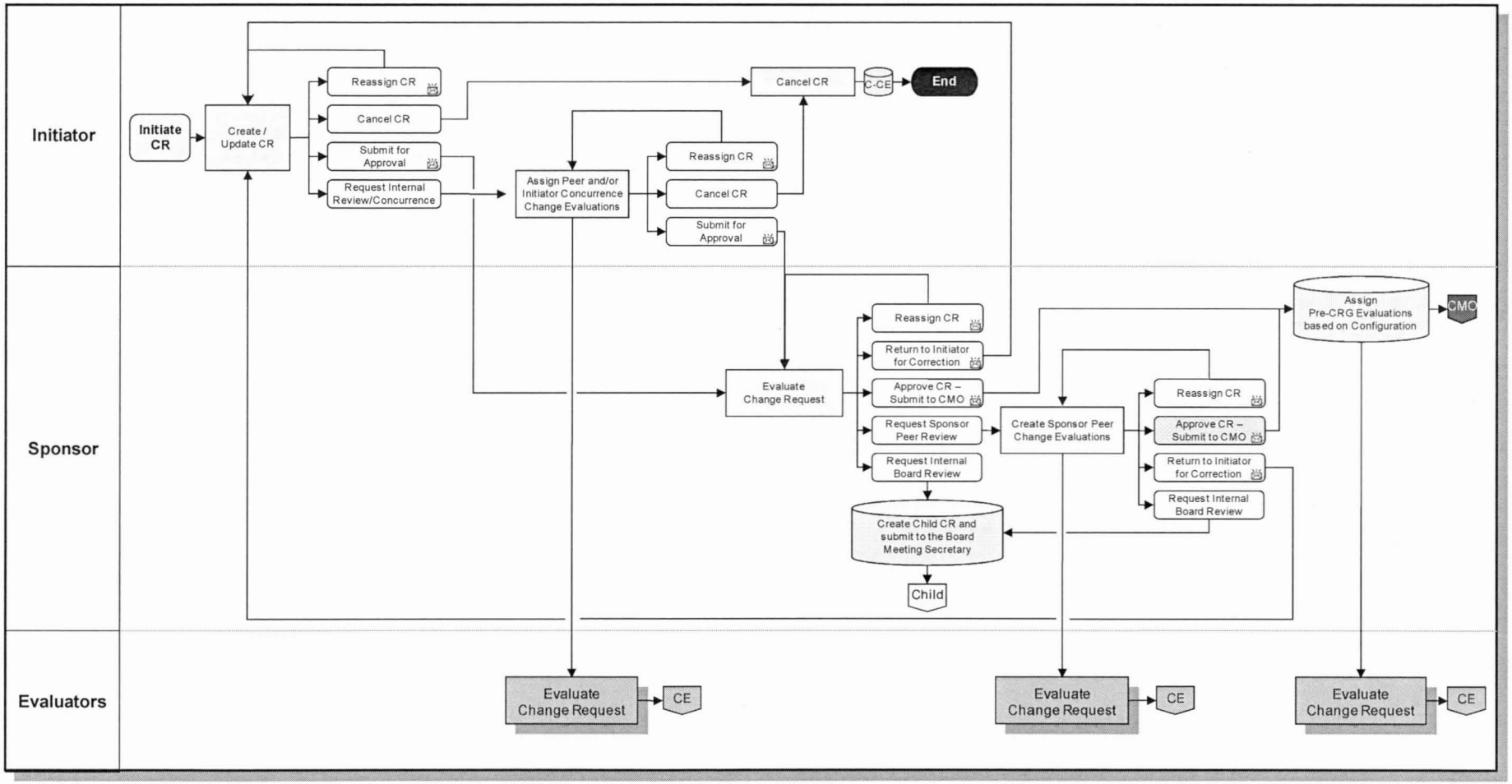
Example 3 – Automated Workflow

- **Workflow is the automation of a business process where information is passed from one state to another for action in accordance with a set of pre-defined business rules**
  - **Configurable workflow design includes a current workflow step, potential next step(s), assignee information, e-mail notifications, and automated process activities**
  - **Workflow is accomplished by the user selecting the next appropriate step from a list of pre-defined\configured options and the system performing the associated activities**



United Space Alliance

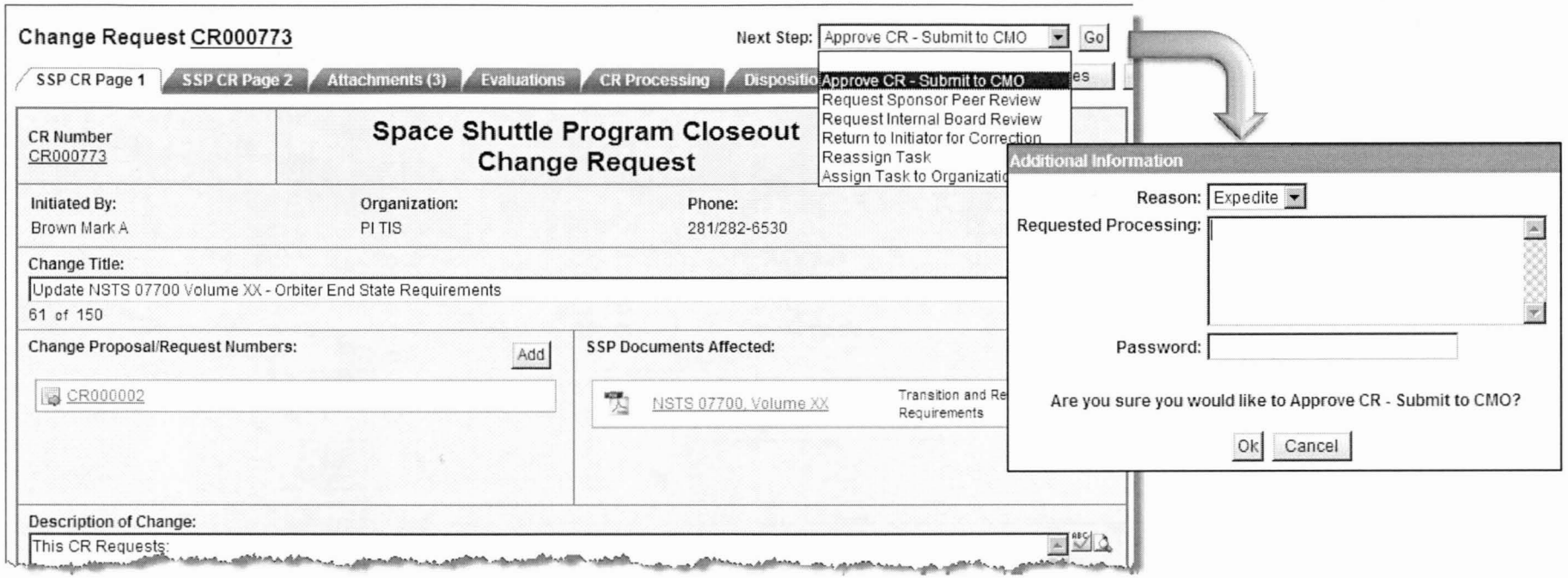
Example 3 – Sample Workflow



Current Step | Next Step Options | Step Event Functions | Spawned Workflow

United Space Alliance

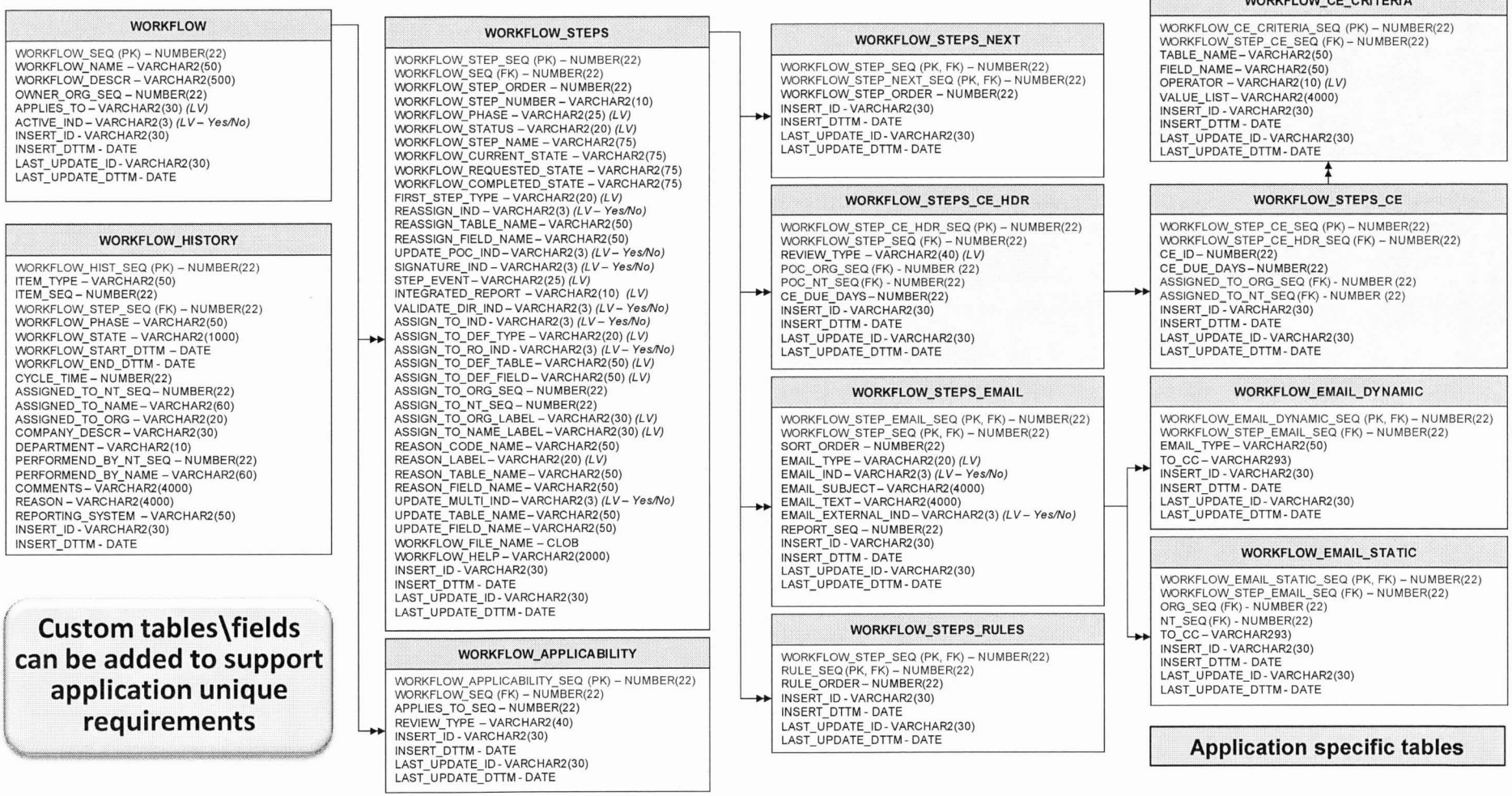
Example 3 – Workflow User Interface



- Based on the workflow configuration, the user is presented with one or more Next Step options
- System determines the processing requirements and opens a pop-up that is dynamically generated based on the selected Next Step
- System runs the Validation Rules, processes the required updates, and notifies the affected users

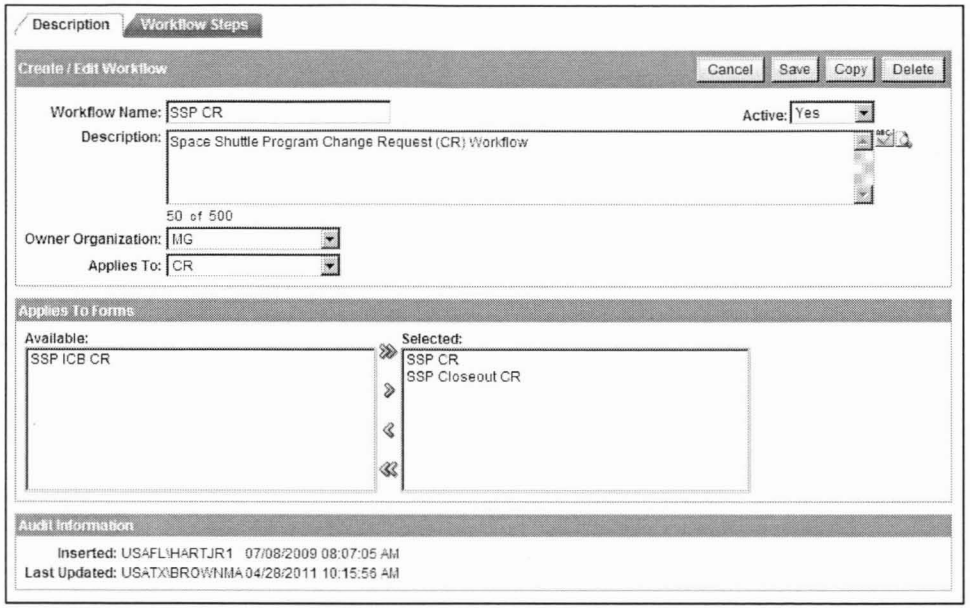
United Space Alliance

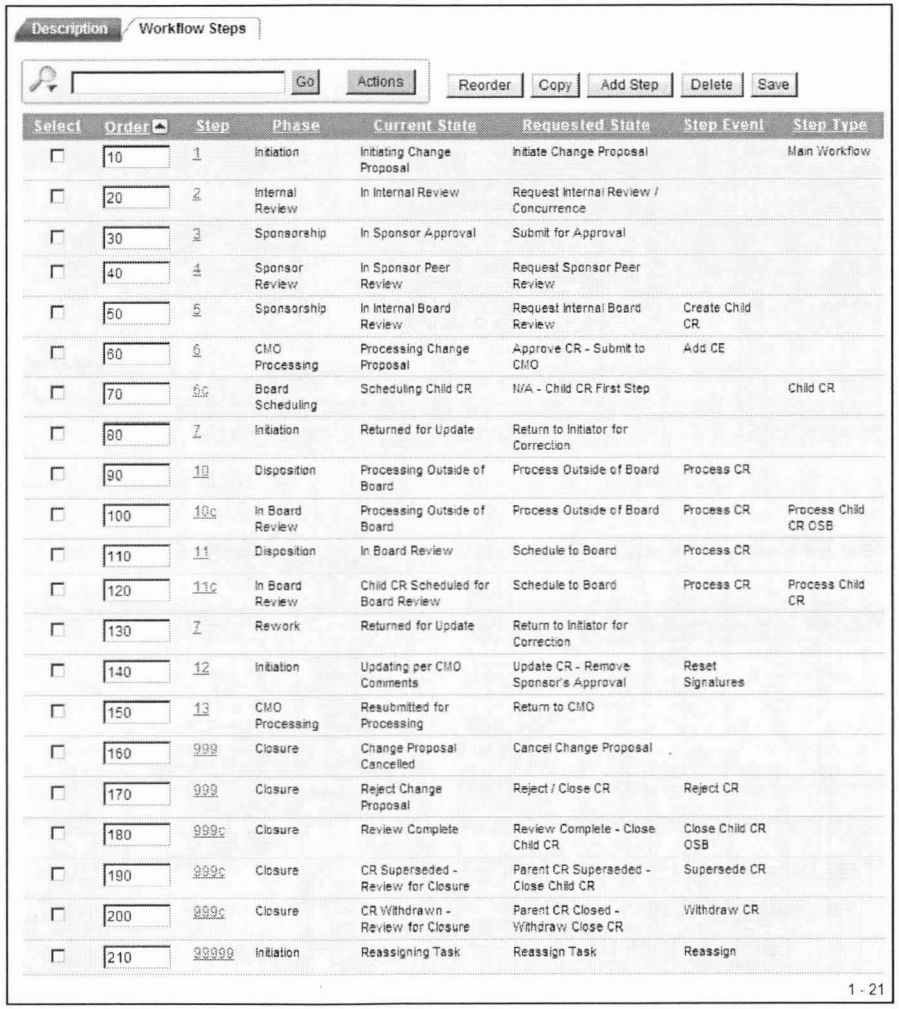
Example 3 – Workflow Design



United Space Alliance

Example 3 – Configuration Design





- **Workflow header identifies the type of workflow and applicability**
- **Workflow steps include the specific processing instructions for each step**
  - **Workflow Assignments**
  - **Step Event Processing**
  - **Dynamic Field Updates**
  - **E-Mail Notifications**
  - **Report Generation**

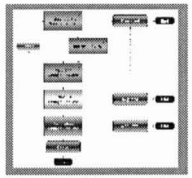
United Space Alliance

Example 3 – Configuration Design

Workflow Step | Next Steps | Add CEs | Data Validation | E-Mail

Create/Edit Workflow Step — Copy | Delete | Cancel | Save

Step Name: Approved - Submit to CMO
Workflow Phase: CMO Processing Workflow Status: Draft
Step Number: 6 Sort Order: 60 Step Type: - Select -
Current State: Processing Change Proposal
Requested State: Approve CR - Submit to CMO
Completed State: CMO Processing Complete
Step Event: Add CE
Printable Version: Create
Include Reassign Option?: Yes Update CE POC?: Yes
Update User in Table: - Select -
Reassign Field Name: - Select -
E-Signature Required?: Yes
Step File Name: Browse...



Assign To Information

Assignment Type: Meeting Secretary Org
Display Assigned To: No
Read Only: - Select -
Table: - Select -
Field: - Select -
Organization: - Select -
Individual: - Select -
Assigned To Org Label:
Assigned To Name Label:

- **Step Events represent PL\SQL procedures that run specific functions when the workflow step is processed**
- **Assigned To Information can be static (read-only) or dynamically derived**
- **Help information for the Next Step options are displayed from the Workflow label**

Response Comments / Rationale

Legal Value Code Name: RECOMMENCED_PROCESSING
Message Label: Requested Processing
Load Response into Table: CMST_CR_PROCESSING_VW
Load Response into Field: RECOMMENCED_PROCESSIN

Workflow Completion Updates (Applied when Step is Selected)

Update Multiple Fields: Yes
Update Table: CMST_CR_PROCESSING_VW
Update Field: Sponsor

Workflow Step Help Description

Source | Style | Format | Font | Size

The CR is approved by the sponsor and submitted to the Configuration Management Office for community review and disposition. An automated e-mail notification is sent to the CMO office and a task will appear in their inbox until the CR is processed.

Additional help is located in the CMST Project Website: https://usa1.usa-spaceops.com/

Audit Information

Inserted: USATX\BROWNMA 07/10/2009 10:09:13AM
Last Updated: USATX\BROWNMA 04/28/2011 10:44:20AM

United Space Alliance

Example 3 – Sample Step Events

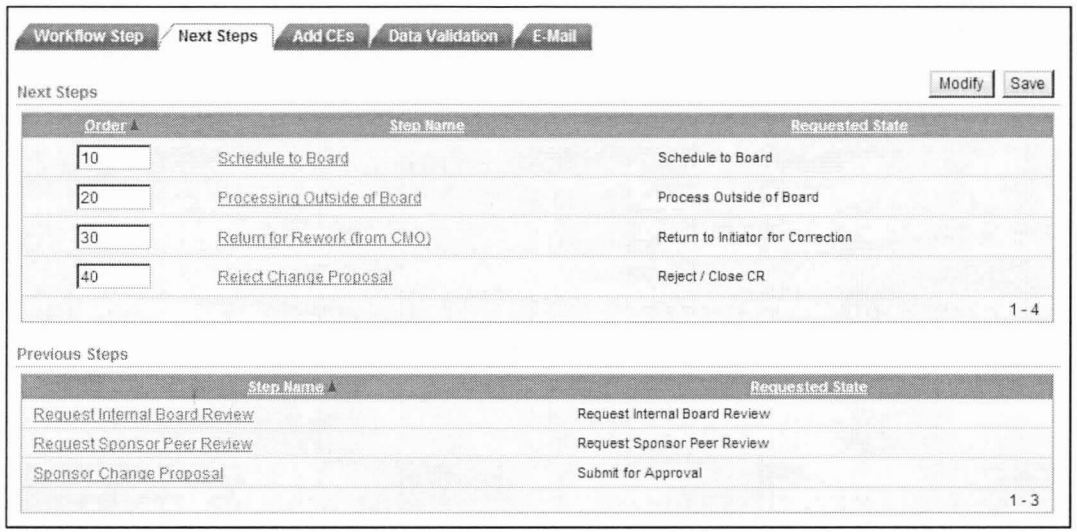
```
-- Step Event
      CASE NVL(p_workflow_step.STEP_EVENT, 'NULL')
        WHEN 'Add CE' THEN
          ADD_CE_FROM_WORKFLOW(
            p_cr_processing_seq => p_cr_processing_seq
           ,p_workflow_step_seq => p_workflow_step.WORKFLOW_STEP_SEQ
           ,p_poc_org => v_assign_to_org
           ,p_poc_user => v_assign_to_user);

        WHEN 'Process CR' THEN
          UPDATE CMST_CR_PROCESSING
             SET CRG_DATE = SYSDATE,
                 CRG_CHAIR_NT_SEQ = p_user.NT_SEQ,
                 CRG_CHAIR_NAME = p_user.NAME
           WHERE CR_PROCESSING_SEQ = p_cr_processing_seq;
          IF v_cr.PROCESSED_OSB_IND = 'Yes' THEN
            DELETE FROM CMST_CR_CRG
             WHERE CR_PROCESSING_SEQ = p_cr_processing_seq;
          ELSE
            FOR crg IN(SELECT CR_CRG_SEQ
                         FROM CMST_CR_CRG
                        WHERE CR_PROCESSING_SEQ = p_cr_processing_seq)
            LOOP
              CMST_AGENDA_PACK.PROCESS_CRG(
                 p_cr_crg_seq => crg.CR_CRG_SEQ
                ,p_cr_seq     => v_cr.CR_SEQ
                ,p_cr_processing_seq => p_cr_processing_seq);
            END LOOP;
          END IF;

        WHEN 'Reset Signatures' THEN
          UPDATE CMST_CR
             SET SPONSOR_ORG_SEQ = NULL,
                 SPONSOR_ORG = NULL,
                 SPONSOR_ORG_DESCR = NULL,
                 SPONSOR_NT_SEQ = NULL,
                 SPONSOR_NAME = NULL,
                 SPONSOR_COMPANY = NULL,
                 SPONSOR_DTTM = NULL
           WHERE CR_SEQ = v_cr.CR_SEQ;
```

"Add CE" creates new Change Evaluations based on the Workflow CE configuration

"Process CR" updates the Change Review Group Information and adds the CR to the Review Board Agenda

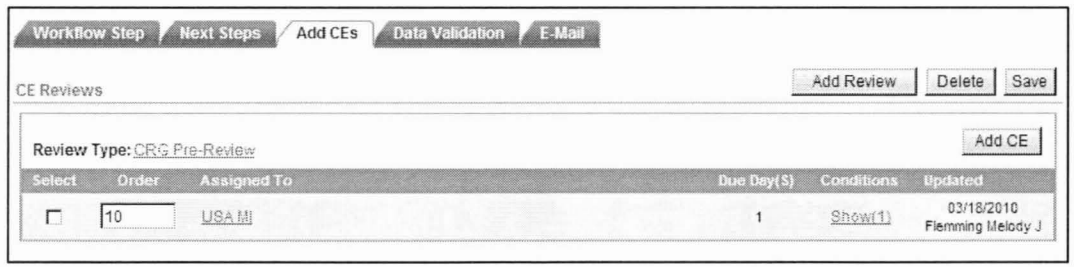
"Reset Signatures" removes the applied Signature Information

Example 3 – Configuration Design

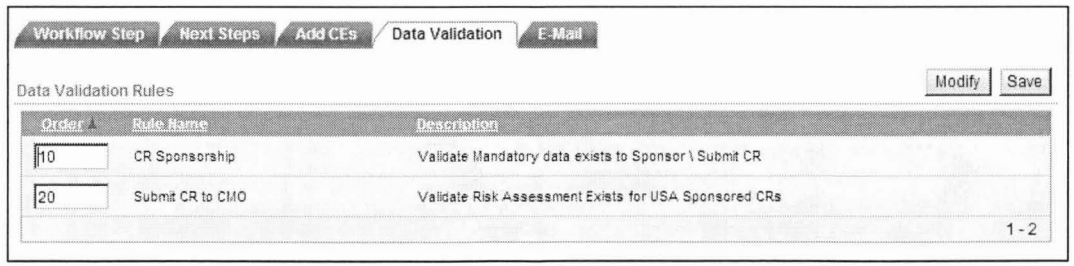


Workflow steps are linked together to create an end-to-end integrated process

Linked steps can spawn other workflow sub-processes



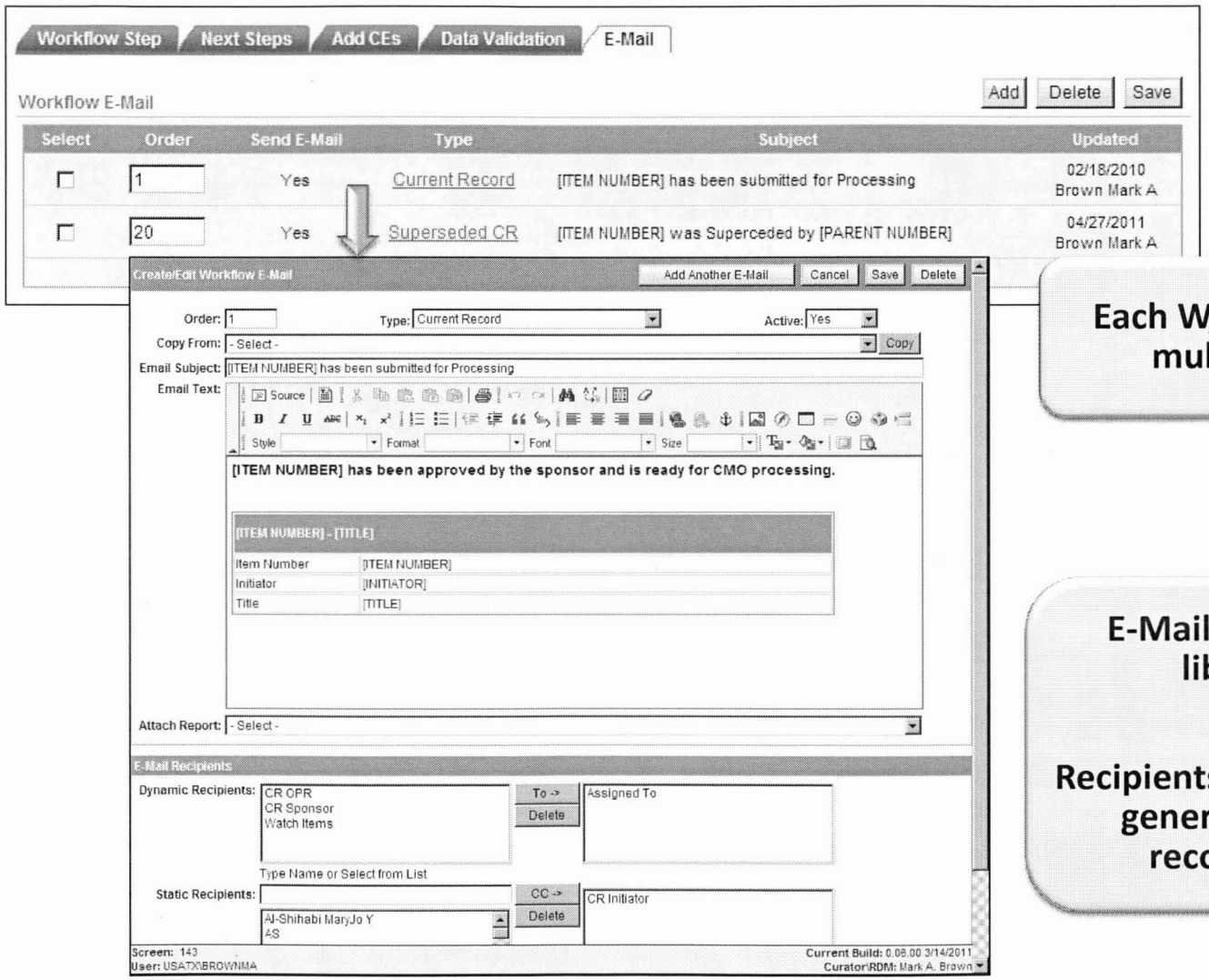
Automatically generate new Tasks



Multiple Validation Rules can be linked to each workflow steps

United Space Alliance

Example 3 – Configuration Design



Each Workflow Step can generate multiple e-mail notifications

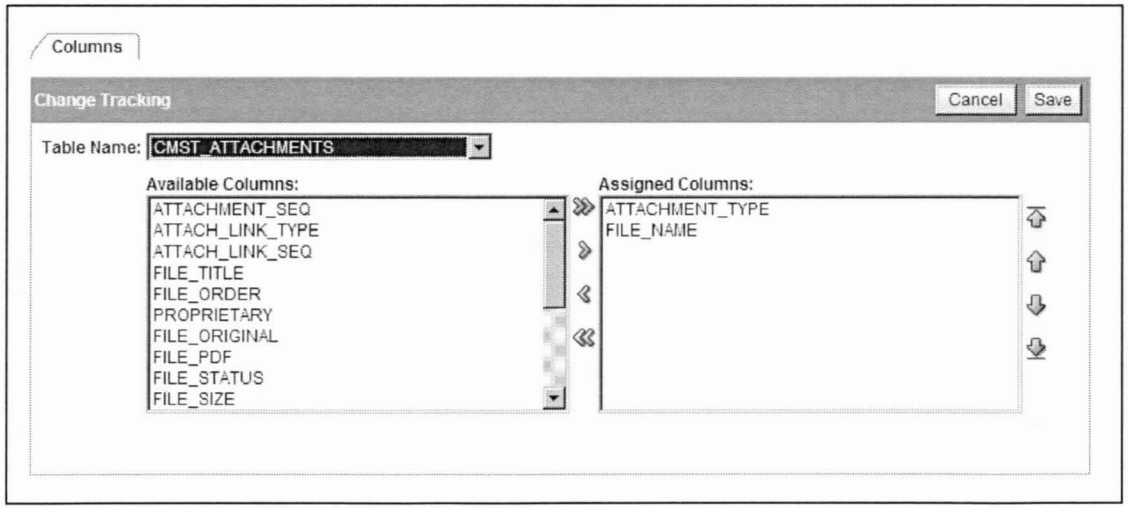
E-Mails are configured using a library of dynamic tags

Recipients can either be dynamically generated based on the linked record or defined statically

United Space Alliance

Example 4 – Audit Tracking Design

- Audit tracking at the field level is a balance between business need and system performance
- Configurable audit tracking allows business owner to decide which fields require traceability and alter as business processes change
- Database trigger is built based on Field Selections dynamically



AUDIT_TABLES
AUDIT_TABLE_SEQ (PK) – VARCHAR2(100)
TABLE_NAME – VARCHAR2(30)
COLUMN_NAME – VARCHAR2(30)
DISPLAY_ORDER – NUMBER(22)
TRIGGER_COMPILED - DATE
INSERT_ID - VARCHAR2(30)
INSERT_DTTM - DATE
LAST_UPDATE_ID - VARCHAR2(30)
LAST_UPDATE_DTTM - DATE

AUDIT_TEMPLATE
AUDIT_SEQ (PK) – NUMBER(22)
AUDIT_DATE – DATE
AUDIT_ACTION – VARCHAR2(15)
COLUMN_NAME – VARCHAR2(30)
NEW_VALUE – VARCHAR2(4000)
OLD_VALUE – VARCHAR2(4000)
DESCRIPTOR – VARCHAR2(100)
REPORTING_SYSTEM – VARCHAR2(50)
LAST_UPDATE_ID - VARCHAR2(30)

Example 4 – Dynamic Trigger Generation

Step 1: Find the Audit Columns

```
cursor col_cursor is
    select column_name, rownum
      from CMST_audit_tables
     where table_name = audit_table_name;

  cursor cons_cursor is
    select column_name, position
      from all_cons_columns
     where table_name = audit_table_name
       and  (constraint_name like 'PK_%'  OR constraint_name like '%_PK')
       and owner = 'APPL_CMST'
    order by position;

  cursor CHILD_CONS_CURSOR IS
  SELECT DISTINCT child_cc.COLUMN_NAME --, rownum, child_cc.position
      FROM all_cons_columns child_cc,
           all_cons_columns parent_cc,
           all_tab_columns atc
     WHERE child_cc.OWNER = 'APPL_CMST'
       AND child_cc.table_name = audit_table_name
       AND atc.COLUMN_ID < 4
       AND child_cc.CONSTRAINT_NAME LIKE 'FK_%'
       AND  (PARENT_CC.constraint_name like 'PK_%'  OR PARENT_CC.constraint_name like '%_PK')
       AND parent_cc.column_name = child_cc.COLUMN_NAME
       AND atc.TABLE_NAME = child_cc.TABLE_NAME
       AND atc.COLUMN_NAME = child_cc.COLUMN_NAME
```

Selects Columns identified in the Panel

Finds Primary Key to store with audit records

Finds Foreign Keys associated with record

Step 2: Create the Trigger

```
v_trigger_name := trim(substr('A_AUD_'||audit_table_name,1,30));
  sql_hdr     := 'CREATE OR REPLACE TRIGGER ' ||v_trigger_name || '
                  AFTER UPDATE or DELETE on ' ||audit_table_name || '
                  for each row DECLARE v_descriptor VARCHAR2(100); BEGIN ';
```


USA™
United Space Alliance

Example 4 – Dynamic Trigger Generation

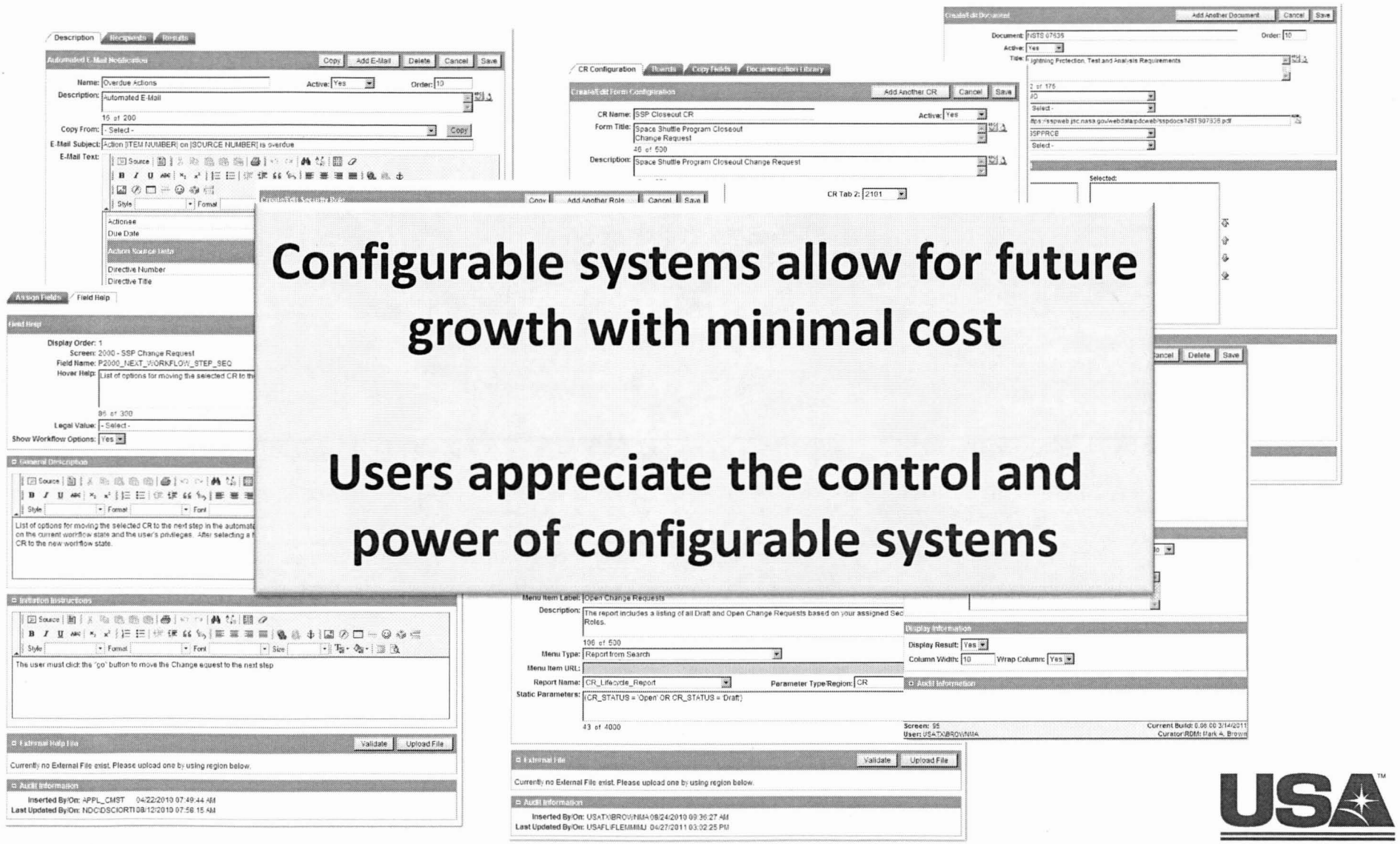
➢ Step 3: Add Dynamic Body

```
:= trigger_body || ' IF UPDATING THEN ';

sql_cols := sql_seq_name||', AUDIT_ACTION, AUDIT_DATE, LAST_UPDATE_ID, COLUMN_NAME,   OLD_VALUE, NEW_VALUE, DESCRIPTOR';
  open cons_cursor;
  loop
    fetch cons_cursor into v_con_col, v_con_cnt;
    exit when cons_cursor%notfound;
      sql_cols := sql_cols||', '||v_con_col;
      v_insert_values_suffix := v_insert_values_suffix||', :old.'||v_con_col;
  end loop;
  close cons_cursor;
open col_cursor;
  loop
    fetch col_cursor into v_col, v_cnt;
    exit when col_cursor%notfound;
    v_if_value_compare := ' IF (nvl(to_char(:new.'||v_col||''),''A'')  <> nvl(to_char(:old.'||v_col||''),''A'')) then ';
  select data_type
      into v_data_type
      from all_tab_columns
      where table_name = audit_table_name
        and column_name = v_col
        AND OWNER = 'APPL_CMST';
  If  v_data_type = 'DATE' then
       sql_valuesI := ' VALUES ('||sql_seq_name||'.NEXTVAL,''U'',sysdate,    nvl(wwv_flow.g_user,user),'''||v_col||''',
to_char(:old.'||v_col||',''DD-Mon-YYYY HH:Mi:SS am''), to_char(:new.'||v_col||',''DD-Mon-YYYY  HH:Mi:SS am''), v_descriptor';
  elsif  v_data_type = 'CLOB' then
      v_if_value_compare := '    IF (nvl(:new.'||v_col||',''A'')  <> nvl(:old.'||v_col||',''A'')) then';
      sql_valuesI := ' VALUES ('||sql_seq_name||'.NEXTVAL,''U'',sysdate, nvl(wwv_flow.g_user,user),'''||v_col||''',:old.'||v_col||',
:new.'||v_col || ', v_descriptor';
  else
      sql_valuesI := ' VALUES ('||sql_seq_name||'.NEXTVAL,''U'',sysdate,
nvl(wwv_flow.g_user,user),'''||v_col||''',to_char(:old.'||v_col||''), to_char(:new.'||v_col||''), v_descriptor';
  end if;
    sql_valuesI := sql_valuesI || v_insert_values_suffix || ')';
    trigger_body := trigger_body || ' ' ||
                    v_if_value_compare || ' ' || --if new value != old_value then
                      v_insert_stmt_prefix || sql_valuesI || '; ' || ' ' ||  --insert into audit table
                    ' end if; ';
  end loop;
  trigger_body := trigger_body || ' end if; '; --end of 'IF UPDATING THEN'
  close col_cursor;
trigger_body1 := trigger_body1 || ' end;';       --end of sql_hdr begin
  execute immediate SQL_HDR || ' ' || trigger_body || ' ' || trigger_body1;
```

Loop through columns and build values statement

USA™
United Space Alliance

No Limits to the Power of Configuration



United Space Alliance